

4th Workshop on Graphical Documentation: UML Style Guidelines

Steve Murphy

Media Design Studio
IBM Toronto Software Laboratory
srmurphy@ca.ibm.com

Scott Tilley

Department of Computer Sciences
Florida Institute of Technology
stilley@cs.fit.edu

Shihong Huang

Department of Computer Science
Florida Atlantic University
shihong@cse.fau.edu

ABSTRACT

The Unified Modeling Language (UML) is the de facto standard for visually representing modern software systems. This workshop will explore UML style guidelines and their effect on the efficacy of UML diagrams in the context of graphical program documentation. This work is part of an ongoing research program focused on assessing system redocumentation techniques. The workshop is a sequel to workshops held at SIGDOC 2001, SIGDOC 2002, and IWPC 2003.

Categories and Subject Descriptors

D.2.7 [Software Engineering]: Maintenance, and Enhancement – Documentation, Reverse Engineering

General Terms

Documentation, Human Factors

Keywords

Documentation, program understanding, visualization

1. MOTIVATION

Graphical representations of software artifacts are often advocated as an effective means of aiding program understanding. There are many factors that can affect the suitability and efficacy of the graphical documentation in this context. For example, the layout algorithm used, edge routing, focal points, node/edge colors and shapes, etc.

For traditional prose, references such as *The Chicago Manual of Style* [7] have served the technical communication community very well. Such reference books provide proven and well-founded guidelines for crafting quality textual documentation. There is little doubt that following the conventions advocated by such trusted sources will result in better technical writing.

Unfortunately, there are no such agreed-upon guidelines for graphical documentation. Without established guidelines, graphical documentation is often haphazardly produced, relying on individual preferences instead of industry standards for quality policies. This includes Unified Modeling Language (UML) diagrams [2]. UML is the de facto standard for visually representing modern software systems, and is rapidly becoming the predominant form of graphical documentation for programmers. The resultant documentation is difficult to objectively assess according to attributes such as correctness, completeness, and effectiveness.

The recent book *The Elements of UML Style* by Scott Ambler [1] offers one starting point for graphical documentation. Ambler speaks at length about the importance of layout issues as a matter of good diagramming style. He offers 236 guidelines on the use of all types of UML Version 1.x diagrams. However, the book's focus is on UML for software design and development. It is unclear whether or not the guidelines are equally applicable to program documentation.

The paper *Designing UML Diagrams for Technical Documentation* [4], published in the ACM SIGDOC 2003 Proceedings, provides practical guidelines for the IBM Toronto Lab technical writers. The guidelines state the roles of the three stakeholders (developer, writer, and graphic designer) in the creation and implementation of usable UML diagrams for technical documentation.

Three recent workshops addressed the issue of graphical documentation in this context. At the SIGDOC 2001 workshop "Documentation for Software Engineers: What is Needed to Aid System Understanding?" one of the questions raised was what format(s) should documentation take to most effectively aid program understanding: textual or graphical [6]. At SIGDOC 2002, the workshop "Graphical Documentation for Programmers" focused on the suitability of three classes of graphical documentation (static, interactive, and editable) for selected program understanding tasks that are representative of common

software maintenance and evolution scenarios [5]. At IWPC 2003, the workshop “Graphical Documentation for Programmers: Assessing the Efficacy of UML Diagrams for Program Understanding” discussed techniques to evaluate the relative efficacy of UML diagrams versus traditional text, and conducted a limited-scope experiment along these lines. [3].

The GDOC 4 workshop continues the discussion at SIGDOC 2004, but will explore UML style guidelines (such as Ambler’s and IBM’s) and their effect on UML diagrams in the context of graphical program documentation.

2. STRUCTURE

The purpose of the workshop is to foster the exchange of ideas and information in an informal setting, but with some boundaries placed on topics and time to ensure that the workshop stays on schedule.

The workshop will have a mix of short presentations and structured discussions. The workshop will begin with a short summary of the previous workshops, so that participants in this year’s workshop have a better understanding of the thread of inquiry that this workshop series represents. The remainder of the workshop will involve critical analysis of representative style guidelines for the UML. The analysis will be from the point of view of efficacy in aiding program documentation.

Relevant results from this workshop will be summarized and published in a suitable forum for public dissemination.

3. PARTICIPANTS

Anyone who attends the SIGDOC 2004 conference is welcome to participate in the workshop. There is no formal prerequisite to participate in the workshop. Although prior knowledge of the UML would be beneficial, it is not absolutely needed. No position paper or presentation is needed. The only requirement is a willingness to actively participate in the workshop discussions.

REFERENCES

- [1] Ambler, S. *The Elements of UML Style*. Cambridge University Press, 2003.
- [2] Object Management Group (OMG). “UML Resource Page”. Online at <http://www.omg.org/uml>.
- [3] Huang, S. and Tilley, S. “Workshop on Graphical Documentation for Programmers: Assessing the Efficacy of UML Diagrams for Program Understanding.” Held in conjunction with *The 11th International Workshop on*

Program Comprehension (IWPC 2003: May 10, 2003; Portland, OR).

- [4] MacKinnon, N. and Murphy, S. “Designing UML Diagrams for Technical Documentation.” *Proceedings of the 21st Annual International Conference on Design of Communication* (SIGDOC 2003: October 12-15, 2003; San Francisco, CA), pp. 105-112. ACM Press: New York, NY, 2003.
- [5] Tilley, S. and Wong, K. “Workshop on Graphical Documentation for Programmers.” Held in conjunction with *The 20th Annual International Conference on Systems Documentation* (SIGDOC 2002: October 21, 2002; Toronto, Canada).
- [6] Tilley, S.; Smith, D.; and Thomas, B. “Documentation for Software Engineers: What is Needed to Aid System Understanding?” Held in conjunction with *The 19th Annual International Conference on Systems Documentation* (SIGDOC 2001: Santa Fe, NM; October 21-24, 2001).
- [7] University of Chicago Press Staff (editors). *The Chicago Manual of Style* (15th Edition). University of Chicago Press, 2003.

ORGANIZERS

Steve Murphy is a media designer at the IBM Toronto Software Laboratory. He leads a team of designers that create graphical support for Information Development writers. He has an honors degree in Fine Arts at the University of Ottawa and a high honors diploma in Illustration at the Sheridan College, Oakville.



Scott Tilley is an Associate Professor in the Department of Computer Sciences at the Florida Institute of Technology. He has a Ph.D. from the University of Victoria. He is the General Chair for ACM SIGDOC 2004, Chair of the Steering Committee for the IEEE Web Site Evolution (WSE) series of events, and the current President of ACM SIGDOC.



Shihong Huang is an Assistant Professor in the Department of Computer Science at Florida Atlantic University. She has a Ph.D. from the University of California, Riverside. She is Chair of Asian Relations for ICSM 2005, and the Program Chair for ACM SIGDOC 2004.

